Методичка 5.3 - Патчинг

Патчинг

В прошлом видео мы выполнили реверс-инжиниринг программы prog-4.1, нашли функцию, которая осуществляет проверку ключа, восстановили алгоритм проверки и смогли создать ключ, который успешно активировал программу.

Давайте представим, что мы не знаем алгоритма проверки ключа или не смогли его восстановить, но нам очень хочется получить рабочий вариант программы.

Суть метода патчинга заключается в том, чтобы найти такой код в программе от которого зависит результат проверки ключа. Как правило, это несколько инструкций, которые выполняют сравнение результата вызова функции со значением, которое заранее известно. Такое место в коде называют триггером. В этом месте в коде принимается принципиальное решение о том, как программа будет себя вести далее.

Для того, чтобы повлиять на проверку ключа, нам необходимо найти такое место и внести соответствующие изменения.

Давайте откроем программу prog-4.1 под отладчиком и перейдем к функции, где программа печатает строку “License key is incorrect.”. Зададим в качестве аргумента произвольный ключ (blabla).

Поставим точку останова сразу после вызова функции проверки ключа.

Мы видим, что далее проверяется значение, которое вернула функция. Значение кладется в регистр ECX, происходит проверка на ноль и если не ноль, то проверка успешно пройдена.

Какие есть варианты патчинга?

1. Сделать так, чтобы функция возвращала всегда 1.
2. Сделать так, чтобы в регистр ECX всегда попадала 1 перед проверкой.
3. Изменить условный переход JNZ на безусловный JMP.

Все эти варианты должны работать. Но самый простой вариант, как можно активировать программу, перед инструкцией JNZ изменить состояние флага ZF. Давайте сделаем это.

Доходим до инструкции JNZ. Кликаем правой кнопкой на флаге ZF - Reset. И видим в консоли надпись - “The program has been activated!”.

Проверяем работу программы в командной строке.

> prog-4.1.exe 1 2

Result: 3

Соответствующие изменения были сделаны в реестре и программа считается активированной.

То, что мы сейчас сделали нельзя сохранить в файле. Для того, чтобы изменения можно было сохранить, нужно изменять инструкции в коде программы.

Давайте удалим записи с реестра, которые сделала программа и восстановим первоначальное состояние.

Открывает regedit и удаляем ветку (HKEY_CURRENT_USER\Software\GeekBrains\Prog4).

Мы видим инструкцию MOVZX ECX, AL которая загружает значение в регистре AL в регистр ECX. Она занимает 3 байта. Во время патчинг нельзя менять количество байт, иначе мы поедут смещения и мы превратим программу в кирпич.

Итак, у нас есть три байта и нам нужно записать значение 1 в регистр ECX. Сразу напрашивается инструкция MOV ECX, 1.

Давайте найдем похожую инструкцию в коде нашей программы и посмотри ее опкод.

Ctrl + F (MOV ECX, 1) - ничего не найдено. Такой инструкции в коде программы нет. Давайте попробуем найти не с единицей, а с двойкой.

Ctrl + F (MOV ECX, 2). Такая инструкция есть. Ее опкод b9 02000000 и такая инструкция нам не подойдет, так как она занимает 5 байт. Первый байт это инструкция и 4 байта это единица.

Возвращаемся к нашему коду. Кликаем правой кнопкой - Go to - Origin.

В такие моменты нужно проявить фантазию и придумать, как за 3 байта получить значение 1 в регистре ECX.

Я придумал следующее. Первая инструкция XOR ECX, ECX, ее опкод 33C9 и составляет 2 байта, а вторая это инструкция INC ECX, ее опкод 41 и составляет 1 байт. В сумме это 3 байта и нам это подходит.

Давайте проверим.

Кликаем правой кнопкой на инструкции Binary - Edit (33C941).

Видим, что код изменился. Давайте выполним эти инструкции. Мы прошли проверку ключа.

Это вариант уже может быть сохранен на диск.

Кликаем правой кнопкой на любое место в коде Copy to executable - All modifications - Copy All.

Появляется новое окно, кликаем на коде правой кнопкой и Save file. Указываем имя новой программы prog-4.1-any-key.exe.

Проверяем работу программы в командной строке.

>prog-4.1-any-key.exe qwerty

The program has been activated!

Отлично! Программа теперь активируется с любым ключом.

А теперь давайте усложним задачу и сделаем так, чтобы программа вообще не требовала активации.

Возвращаемся в отладчик. Откатываем изменения.

Окно Patches - Restore Original Code.

Далее по коду мы видим строку - “The number of starts has expired.”, которая оповещает нас о том, что количество запусков программы исчерпано. Это происходит тогда когда, значение параметра Counter равно 0.

Будет ли напечатана эта строки или нет завист от функции 0x00401300, которая, видимо, проверяет счетчик.

Мы точно не знаем, что делает эта функция, поэтому просто убрать вызов этой функции будет неправильно.

Мы пойдем по-другому пути. Сразу после вызова функции мы видим условный переход JNZ. Именно от него зависит, попадем мы на код, которая завершает работу программы или нет.

Нам важно, чтобы этот переход осуществлялся всегда. Заменим условный переход JNZ на безусловный JMP.

Кликаем правой кнопкой на инструкции Binary - Edit (EB14).

Видим, что код изменился.

Давайте изменим состояние счетчика в реестре на ноль, чтобы спровоцировать состояние, когда количество попыток запуска закончится.

Counter - 0.

Давайте выполним эти инструкции. Видим в консоли, что программа успешно отработала, несмотря на то, что счетчик равен 0. Мы прошли проверку ключа.

Это вариант уже может быть сохранен на диск.

Кликаем правой кнопкой на любое место в коде Copy to executable - All modifications - Copy All.

Появляется новое окно, кликаем на коде правой кнопкой и Save file. Указываем имя новой программы prog-4.1-cracked.exe.

Проверяем работу программы в командной строке.

>prog-4.1-cracked.exe 1 2

Result: 3

Как мы видим, программа работает несмотря на то, что счетчик равен нулю.